To convert from a number into a scale/mantissa floating point code with R_s scale (exponent) bits and R_m mantissa bits. s represents the sign bit (0 = positive, 1 = negative) which is the most significant bit of the mantissa.

I. Quantize the number as an R_u-bit uniform quantization code where $R_u = 2^{R_s} - 1 + Rm$.

II. Count the number of leading zeros in the resulting uniform quantization code, excluding the sign bit, s. If the number of leading zeros is less than $2^{R_s} - 1$, then set the scale equal to the number of leading zeros; otherwise, set the scale equal to $2^{R_s} - 1$.

III. If the scale is equal to $2^{R_s} - 1$, then set the first mantissa bit equal to s, and set the remaining $R_m - 1$ bits equal to the bits following the $2^{R_s} - 1$ leading zeros in —code—; otherwise, set the first mantissa bit equal to s, and set the remaining $R_m - 1$ bits equal to the bits following the leading zeros, $o$mitting the leading one.